INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Complexité de l'élection

Christian LAVAULT

N° 2118
Décembre 1993

PROGRAMME 1

Architectures parallèles,
bases de données,
réseaux et systèmes distribués

Complexité de l'élection

Méthodes d'analyse de complexité

d'algorithmes distribués d'élection

Christian Lavault*

Programme 1 — Architectures parallèles, bases de données, réseaux
et systèmes distribués
Projet MODEL

Rapport de recherche n°2118 — 27 décembre 1993 — 34 pages

Résumé : L'élection sur les anneaux a déjà donné lieu à quantités de recherches, tant dans le cas où les processus sont tous distingués par leurs identités « anneaux avec identités ») que dans celui où ils sont sans identités, indiscernables (« anneaux anonymes »). On sait également différencier les classes d'anneaux selon des critères pertinents du point de vue de la complexité en communication des algorithmes, *i.e.* anneaux synchrones ou asynchrones, uni- ou bidirectionnels, mais aussi anneaux avec ou sans information structurelle (« sens général de la direction », connaissances exacte ou approchée de la taille de l'anneau, etc.) : tout types de connaissance à même d'améliorer considérablement la complexité en communication de bien des algorithmes distribués. Cette synthèse tente de montrer que presque tous les problèmes fondamentaux se posant en algorithmique distribuée apparaissent déjà clairement dans le simple cas de l'election sur des anneaux, et ce, tout particulièrement en ce qui concerne les performances des algorithmes distribués. Elle constitue également une tentative pour faire le point sur les diverses méthodes d'analyse de complexité (au pire et en moyenne) des algorithmes d'élection.

Mots-clé : Système distribué, algorithme distribué, électi on, réseaux en anneau, complexité en communication/en temps, analyse de complexité au pire/en moyenne, optimalité, réseaux avec identités, réseaux anonymes, algorithmes probabilistes.

A preliminary version of the report is to appear in the proceedings of OPOPAC (*International Workshop On Principles Of Parallel Computing*)

*Email : lavault@irisa.fr

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : (33) 99 84 71 00 – Télécopie : (33) 99 38 38 32

Complexity Analysis of Election Algorithms

Abstract: Electing a leader process by a distributed algorithm has received a great deal of attention, both in the case when processes are distinguished by unique identities ("named networks" or "id-based networks") and the case when processes are identical ("anonymous networks" or "nameless networks"). It has also become quite clear that a very important role is played by those factors which can be termed as Structural Information, *i.e.* a priori knowledge available to the processes regarding the structure and characteristics of the system. Such a knowledge can drastically reduce the communication complexity of several distributed algorithms. In the present paper, we try and review some of the classical and more recent solutions for both cases on a ring configuration, since it is now clear that the basic issues already arise in networks with a ring-based topology. Synthetizing and taking stock of the various methods at hand in the worst/average-case analysis of distributed leader finding algorithms

Key-words: Distributed System, Distributed Algorithms, Leader Election, Ring-Based Networks, Communication/Time Complexity, Worst/Average-case Complexity Analysis, Optimal Algorithms, Named Networks, Anonymous Networks, Randomized Algorithms.

Table des matières

1. Introduction

Un des paradigmes de l'informatique distribuée est de *briser la symétrie* des processus en octroyant à l'un d'entre eux un *privilège*. C'est le cas du problème *général* de l'*exclusion mutuelle* et c'est aussi le cas de *l'élection*, où un processus unique est privilégié grâce à un choix distribué de tous les processus coopérant au sein d'un système distribué $\mathcal{S}$. Ce processus une fois élu, il joue ensuite un rôle de coordination et de contrôle dans le système $\mathcal{S}$.

Le processus d'élection dans un système distribué $\mathcal{S}$ se caractérise par une transformation de la configuration de $\mathcal{S}$. Dans une configuration initiale du système, tous les processus actifs sont dans le même état (*candidat*), dans une configuration finale, un seul processus est dans un état prédéterminé (*élu*) et tous les autres dans un état prédéterminé différent du précédent (*battu*).

Dans la synthèse qui suit, nous ne considérerons que deux grandes catégories d'anneaux correspondant à deux « hypothèses de symétrie » distinctes sur les processus :

L'anneau avec identités. Ce type d'anneau est défini par le modèle « standard » de système distribué $\mathcal{S}$: chaque processus de $\mathcal{S}$ y est *distingué* par l'attribution d'une identité unique, distincte des autres (*cf.* § 2.1) et connue de lui seul.

L'anneau anonyme. On dit qu'un anneau est *anonyme* si les processus du système distribué $\mathcal{S}$ y sont *symétriques* du point de vue de l'exécution algorithmique, mais ne possèdent pas d'identités (distinctes) : ils sont alors *indiscernables*.

Le chapitre 3 est consacré à un exemple d'analyse de complexité et d'optimalité (en communication) d'un algorithme distribué d'élection sur un anneau unidirectionnel asynchrone *avec identités* : l'élection consiste ici à briser la symétrie par comparaisons d'identités. Le chapitre 4 propose deux analyses de complexité d'algorithmes distribués d'élection sur des anneaux unidirectionnels *anonymes* ; l'élection consiste ici encore à briser la symétrie, mais par comparaison d'entiers tirés aléatoirement : les algorithmes sont alors probabilistes.

Dans ces exemples, nous nous sommes limités à l'analyse de la complexité en communication et en temps des algorithmes déterministes ou probabilistes, au pire et en moyenne — selon les définitions de ces mesures. Toutes ces analyses sont uniquement mathématiques et les algorithmes n'y sont présentés... *que* de ce point de vue.

2. Système distribué et algorithmes distribués

Ce chapitre se présente comme une introduction rapide et très générale au modèle du système distribué et à la notion d'algorithme distribué ; il ne fait que mettre en place les hypothèses et propriétés nécessaires.

2.1. *Le modèle*

Le modèle standard de *système distribué* (ou *réparti*) $\mathcal{S}$ considéré dans la suite est une structure logicielle et matérielle distribuée en un *réseau point à point asynchrone de processus séquentiels communicants*. Les constituants du réseau d'interconnexion sont les *sites* et le *système de communication* établi entre eux. Cette structure est modélisée par un graphe connexe, simple et symétrique, $G = (P,L)$, où P est l'ensemble fini des sites de $\mathcal{S}$ et L celui des lignes de communication de $\mathcal{S}$; dans toute la suite, $|P| = n$ et $|L| = m$. Chaque site possède une *mémoire locale non partagée* de capacité bornée c, et *au moins un processeur* ; l'échange d'information entre sites s'effectue *uniquement par messages*. Les autres caractères distinctifs d'un tel système distribué *asynchrone* $\mathcal{S}$ sont *l'absence de toute mémoire partagée et d'horloge globale et l'absence de toute variable ou état global* qui serait perceptible par les composants de $\mathcal{S}$ à un « instant » donné. Les seules classes d'événements perçus par un processus quelconque sont donc soit les événements produits par le processus lui-même de façon « spontanée » (par exemple l'envoi d'un message), soit des événements causés par d'autres processus (par exemple la réception d'un ou plusieurs message émis par un ou plusieurs autres processus).

Par ailleurs, dans la première partie (§ 3), chaque site ou processus de $\mathcal{S}$ possède un numéro d'identification *unique*, son *identité*. Soit I un ensemble non vide quelconque, fini ou infini, les identités des processus sont tirées du sous-ensemble $D(I)$ des suites *finies*, non vides d'éléments *distincts* de I : l'ensemble des identités dans $\mathcal{S}$ est ainsi muni d'un ordre total strict, et, dans ce modèle, les identités de tous les processus sont donc, par hypothèse, bien distinctes. La situation où les processus, en tout ou partie, possèdent des identités non distinctes — c'est-à-dire de fait *aucune identité* — est abordée dans la seconde partie (§ 4) ; dans ce cas le réseau est dit *anonyme*, les processus y sont *identiques* ou *indiscernables*.

2.2. *Les algorithmes distribués*

Un *algorithme distribué* A sur un système distribué $\mathcal{S}$ n'est autre que la caractérisation des *transitions locales* à réaliser séquentiellement par chaque processus de $\mathcal{S}$ à la réception d'un message, ce processus étant dans un état déterminé. A peut être vu comme une collection de processus qui, par échange d'information, participent à la réalisation d'un but commun tout en conservant leur autonomie. Chaque processus peut alors être défini comme une *suite d'événements*, un ensemble fini d'états et un ensemble fini de *transitions atomiques* entre événements ainsi schématisés :

$$\langle\ \text{état}_i\ ,\ \langle\text{événement}\rangle_j\ \rangle \longrightarrow \langle\ \text{état}_{i+1}\ ,\ \langle\text{événement}\rangle_{j+1}\ \rangle,$$

où ⟨événement⟩$_j$ est une $j^{\text{ième}}$ *action atomique* du processus. Un *événement interne* produit seulement un changement d'état, un *événement d'envoi* provoque l'envoi d'un message asynchrone, et un *événement de réception* donne lieu à la fois à la réception d'un message et à la mise à jour de l'état local selon les valeurs du message.

On ne peut donc considérer le concept d'algorithme distribué comme une simple variante de celui d'algorithme séquentiel ; un algorithme distribué n'est *pas* un algorithme séquentiel considéré dans un environnement distribué multiprocesseur. Un algorithme distribué A est défini *sur un système distribué* $\mathcal{S}$, lequel suppose des processus et des lignes de communication, un état local et une mémoire locale, etc. A est ainsi d'emblée une construction où la communication joue un rôle central et où, par conséquent, la causalité entre événements, la connaissance locale, la connaissance commune et l'apprentissage des processus, etc. sont des notions à la fois fondamentales et absolument spécifiques à l'environnement distribué.

Depuis maintenant une quinzaine d'années [LeLann-77], le modèle de système distribué du § 2.1 et la notion d'algorithme distribué du § 2.2 sont presque unanimement utilisés dans la littérature spécialisée, à quelques variantes près, qui sont souvent conçues pour les besoins des analyses de complexité : complexité en temps dans un réseau asynchrone, complexité en moyenne, calculs de bornes inférieures, etc. (*cf.* [LAVAULT-90], [BODLAENDER-91], etc.)

3. L'élection sur un anneau avec identités — l'algorithme de Chang-Roberts

Le problème de l'élection dans un système distribué standard défini comme au chapitre 2.1 est considéré, au même titre que celui de l'exclusion mutuelle, comme l'un des paradigmes de l'informatique distribuée. Depuis la fin des années 70, de très nombreux protocoles d'élection ont été conçus en utilisant divers modèles de systèmes distribués et dans des topologies d'anneaux avec identités (non anonymes) très diverses, l'anneau unidirectionnel ou bidirectionnel avec ou sans le sens de la direction, synchrone ou asynchrone, etc.

Étant donné un ensemble I quelconque, on définit le sous-ensemble $D(I)$ comme dans le modèle du § 2.1. On suppose, pour simplifier les notations des algorithmes et leur analyse, que les n processus sont numérotés de 1 à n sur l'anneau, c'est-à-dire $P_1, P_2, \ldots, P_n$, dans le sens des aiguille d'une montre. Il s'agit là d'une simple commodité : cette numérotation reste, bien sûr, inconnue des processus et, *a priori*, absolument sans rapport avec leurs identités.

Définition 1 *Soit I un ensemble quelconque d'indices. $D_n(I)$ est le sous-ensemble fini des suites (ou mots) s non vides d'éléments distincts de I de longueur n : $D_n(I) = \{s \in D(I) / |s| = n\}$, où $|s|$ est la longueur de s.*

On dit qu'un anneau est étiqueté par $s = \langle id_1\ id_2 \cdots id_n \rangle \in D_n(I)$ si, et seulement si, pour tout $i \in [n]$, le processus P_i a pour identité id_i.

Jusqu'à présent, l'algorithme de Chang et Roberts de 1979 [CHANG, ROBERTS-79] reste à la fois le plus simple et le plus performant (en nombre moyen de messages) des algorithmes distribués d'élection sur un anneau unidirectionnel asynchrone ; non seulement il est *en moyenne optimal* en nombre de messages, à savoir en $\Theta(n \lg n)$, mais il opère de manière remarquablement efficace et fiable en moyenne : c'est sans aucun doute l'algorithme « à utiliser » pour effectuer une élection sur un anneau.

L'algorithme d'élection de Chang-Roberts est très faciles à analyser dans le pire des cas, aucun outillage mathématique particulier n'est vraiment nécessaire. Pour l'analyse en moyenne, il suffit, comme bien souvent, d'utiliser des résultats de la combinatoire des permutations et des arguments probabilistes ; ces résultats restent cependant très simples ici. En effet, une configuration quelconque de n processus sur un anneau n'est rien d'autre qu'une certaine permutation de leurs n identités prises dans $D_n(I)$, selon la définition 1, c'est-à-dire une permutation dans $\mathfrak{S}_n$ [1] L'algorithme réalise en fait un calcul d'extremum qui, comme beaucoup d'algorithmes distribués d'élection sur des anneaux, utilise le principe de l'*extinction sélective*. Plus difficile, car plus technique, est la démonstration d'optimalité de l'algorithme de Chang-Roberts, en particulier le calcul de la borne inférieure du nombre moyen de messages de la classe ($\mathfrak{E}$) des algorithmes distribués d'élection par calcul d'extremum sur des anneaux asynchrones unidirectionnels.

3.1. *Hypothèses fondamentales*

L'algorithme élit ici le processus d'identité *maximale* et non celui d'identité minimale sur l'anneau. Si ce choix n'a bien sûr aucune influence sur la complexité en message elle-même, il n'influe pas non plus sur la taille maximale des messages, puisque celle-ci reste dans les deux cas en $O(\lg id^*)$, où id^* est l'identité maximale des n processus : au moins un message contenant l'identité maximale doit de toute façon être envoyé une fois pour toute exécution de l'algorithme. En revanche, la complexité moyenne en bits de l'algorithme qui élit le processus d'identité *minimale* est légérement meilleure — à une constante près (*cf.* la remarque 1 en fin de chapitre).

---

[1] Une permutation $\sigma = \langle \sigma_1\ \sigma_2 \cdots \sigma_n \rangle$ d'un ensemble fini E, $|E| = n$, est une bijection de E dans lui-même. Pour simplifier, l'ensemble des permutations de E est noté $\mathfrak{S}_n$, le groupe symétrique de E à $n!$ éléments

Les hypothèses fondamentales sur l'anneau et sur les processus sont celles du §.1.2. Le réseau en anneau considéré est unidirectionnel et les processus n'ont *aucune* connaissance globale, ni le sens général de la direction, ni la taille n de l'anneau, ni même l'identité de chacun de leurs deux voisins.

Du point de vue de l'analyse de la complexité de l'algorithme, on sait que l'anneau étant unidirectionnel et les transmissions le long des lignes de communication de type « FIFO », le nombre de messages échangés ne dépend pas des délais relatifs de transmission. On peut donc faire une dernière hypothèse : les n processus sont supposés actifs conjointement sur l'anneau au temps $t = 1$ et les délais de transmission des messages le long des liens de l'anneau d'*une «unité de temps »*. Si un nombre $a < n$ de processus deviennent actifs, éventuellement chacun à un moment différent, la complexité moyenne en messages de l'algorithme de Chang-Roberts est strictement inférieure, à savoir : $nH_a + O(n)$, où H_a est le $a^{\text{ième}}$ nombre harmonique (a entier) [2]. L'hypothèse de $a = n$ processus actifs conjointement et spontanément est donc logique et habituelle dans l'analyse des algorithmes distribués, en particulier en moyenne.

3.2. *Présentation de l'algorithme*

L'algorithme se déroule en trois étapes,

(i) la détermination du processus d'identité maximale et l'arrêt des autres candidatures par extinction sélective : chaque processus P_i $(1 \leq i \leq n)$ envoie un message contenant sa propre identité, $\langle id_i \rangle$ (id_i élément de $s \in D_n(I)$), dans la même direction sur l'anneau. Tout message $\langle id_i \rangle$ est « éteint » par le premier processus P_j rencontré sur l'anneau possédant une identité id_j telle que $id_j > id_i$. Ainsi, tous les messages $\langle id_i \rangle$ sont éteints le long de l'anneau durant leur transit, à l'exception du message qui contient l'identité maximale, notée id^* dans l'étiquetage $s = \langle id_1\ id_2 \cdots id_n \rangle \in D_n(I)$.

(ii) la fin de l'élection : le processus élu s'identifie lui-même comme tel, il est l'unique processus qui finit par recevoir comme message sa propre identité, $\langle id^* \rangle$. Seul le processus d'identité maximale est dans ce cas puisque les messages des autres processus sont tous éteints sur l'anneau avant d'atteindre leur processus d'origine, ou initiateur.

---

[2] On note $H_n = \sum_{i=1}^{n} 1/i$ le $n^{\text{ième}}$ nombre harmonique, dont le développement asymptotique est $H_n = \ln n + \gamma + 1/2n - 1/12n^2 + O(n^{-4})$ $(n \to \infty)$, où γ est la constante d'Euler : $\gamma = 0{,}577\ldots$ Par ailleurs, dans toute la suite, ln représente le logarithme népérien et lg le logarithme en base deux : $\lg n = \log_2(n)$.

(iii) l'annonce de l'élection, ou « inauguration » : Il reste alors au processus ainsi élu à envoyer $n - 1$ messages de fin de reconnaissance avec son identité, $\langle \mathbf{fin}, id^* \rangle$, aux $n - 1$ autres processus pour terminer correctement l'élection ; tout processus sur l'anneau doit en effet connaître l'élu à la fin du processus d'élection. Les processus entrent alors dans l'état *passif* et le système devient quiescent.

Exemple La figure 1 ci-dessous représente une élection sur un anneau unidirectionnel avec $D_n(I) = \mathfrak{S}([n]) = \mathfrak{S}_n$, où $n = 5$; l'anneau est étiqueté par une permutation de [5] : $\sigma = \langle 3\ 4\ 1\ 2\ 5 \rangle$.

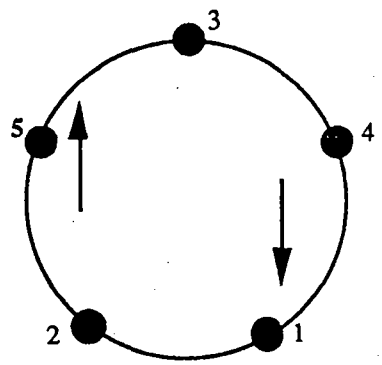

Figure 1. *Election sur un anneau unidirectionnel (Chang-Roberts)*

On peut trouver dans [RAYNAL-85, p. 30], par exemple, la spécification et la preuve de l'algorithme de Chang-Roberts.

3.3. *Analyse de l'algorithme*

Théorème 1 *Soit $D_n(I)$ un sous-ensemble d'identités des processus d'un anneau unidirectionnel asynchrone de taille n. L'algorithme de Chang-Roberts résout le problème de l'élection sur l'anneau en utilisant $\frac{1}{2}n^2 + O(n)$ messages dans le pire des cas et $nH_n = n \ln n + O(n) = 0{,}69\ldots\ n \lg n + O(n)$ messages en moyenne, la taille des messages étant $O(\lg n)$. Il est optimal en moyenne dans la classe $(\mathfrak{E})$. La complexité en temps de l'algorithme est d'au moins $2n - 1$ dans le pire des cas.*

Démonstration. Nous démontrons le théorème 1 dans l'ordre suivant : complexité en messages de l'algorithme dans le pire des cas, en moyenne, optimalité en mes-

sages en moyenne dans ($\bar{\delta}$), comportement de l'algorithme et enfin complexité en temps.

3.3.1. *Complexité en messages de l'algorithme dans le pire des cas*

La configuration extrémale la moins favorable du point de vue du nombre de messages transmis est celle où les identités des processus constituent une permutation $\sigma \in D_n(I)$ dont les éléments sont *décroissants dans le sens de la direction sur l'anneau*, à savoir $\sigma = \langle id_n\ id_{n-1} \cdots id_2\ id_1 \rangle$, où $id_i > id_{i-1}$ $(1 < i \leq n)$. Il y a alors un message $\langle id_1 \rangle$, deux messages $\langle id_2 \rangle$, ..., n messages $\langle id_n \rangle$ envoyés pour l'élection, et $n-1$ messages $\langle \mathbf{fin}, id^* \rangle$ qui « inaugurent » le processus élu d'identité $id_n = id^* = \max_{1 \leq i \leq n} \{id_i\}$.

Dans le pire des cas, le nombre de messages utilisés par l'algorithme est exactement

$$\sum_{i=1}^{n} i + (n-1) = \frac{n(n+1)}{2} + n - 1$$

et la complexité en messages au pire de l'algorithme est donc

$$\frac{1}{2} n^2 + O(n) = \Theta(n^2). \qquad \blacksquare$$

Remarque Jusqu'en 1993, l'algorithme d'élection le plus performant dans le pire des cas sur un anneau unidirectionnel asynchrone est resté celui proposé par Dany Dolev, Mike Klawe et Michael Rodeh en 1982 [DOLEV,KLAWE,RODEH-82] dont la complexité au pire est 1,356... n lg$n$. Ce majorant ($\simeq$ 1,956 n ln$n$ messages) est resté la meilleure complexité connue pendant plus de dix ans, mais en 1993, Lisa Higham et T. Przytycka [HIGHAM,PRZYTYCKA-93] ont amélioré l'algorithme précédent et obtenu un majorant de 1,271... n lg$n$ ($\simeq$ 1,833 n ln$n$) messages.

3.3.2. *Complexité en messages de l'algorithme en moyenne*

La démonstration qui suit est essentiellement probabiliste et permet le calcul de la variance du nombre de messages qui sert ensuite au § 3.3.4. C'est une variante « développée » de la démonstration originale de [CHANG, ROBERTS-79] et de [ROTEM,KORACH,SANTORO-87] (voir aussi [LAVAULT-87], [TEL-94], etc).

Un anneau unidirectionnel étant étiqueté par une permutation $\sigma \in D_n(I)$, supposons que les identités des processus y sont ordonnées en sorte que les $n!$ permutations d'identités soient équiprobables. Il s'agit de calculer le nombre moyen de liens parcourus sur l'anneau avant leur extinction par les messages envoyés au cours

de l'algorithme dans le sens des aiguilles d'une montre. Considérons $a \leq n$ processus actifs. Puisque les identités des processus sont ordonnées de manière équiprobable sur l'anneau, le message envoyé par un processus actif possédant la $j^{\text{ième}}$ plus *grande* identité parcourt *en moyenne n/j* liens sur l'anneau avant d'être éteint par un processus d'identité supérieure. Par conséquent, avec a processus actifs, l'élection est réalisée en utilisant un nombre moyen de messages exactement égal à

$$\sum_{j=1}^{a} n/j = nH_a.$$

Si tous les processus sont actifs ($a = n$), l'élection est réalisée en utilisant un nombre moyen de messages exactement égal à nH_n.

En ajoutant les $n - 1$ messages d'inauguration, la complexité moyenne en messages de l'algorithme de Chang-Roberts est donc

$$nH_n + n - 1 = n \ln n + O(n) = 0{,}69\ldots\ n \lg n + O(n) = \Theta(n \lg n). \ \blacksquare$$

Ce calcul suffirait — et suffit — pour évaluer cette complexité. Mais, nous devons aller un peu plus loin pour calculer la variance du nombre de messages.

Notons M_n la variable aléatoire représentant le nombre total de messages transmis par l'algorithme de Chang-Roberts sur un anneau de taille n, et $E[M_n]$ l'espérance de cette variable aléatoire, *i.e.* le nombre moyen de messages transmis.

Le parcours d'un message envoyé le long des liens d'un anneau unidirectionnel par le processus possédant la $i^{\text{ème}}$ plus *petite* identité (P_i), peut être modélisé par le tirage aléatoire sans remise de $n - 1$ boules contenues dans une urne. Celle-ci contient $n - i$ boules noires et $i - 1$ boules blanches et, dès qu'une boule noire a été tirée, l'expérience prend fin. Pour $1 \leq i \leq n - 1$, le nombre de tirages aléatoires effectués avant de sortir une boule noire est une variable aléatoire T_i qui suit une *loi hypergéométrique négative*, d'espérance $E[T_i] = \dfrac{n}{n - i + 1}$ (voir [JOHNSON, KOTZ-77, p. 84-85]). T_i n'est pas définie pour $i > n - 1$ et, pour tout $i \in [n - 1]$, $E[T_i]$ correspond au nombre de liens parcourus *en moyenne* sur l'anneau par le message d'initiateur P_i avant son extinction par un processus d'identité supérieure. Pour $i > n - 1$, la contribution à la moyenne du message d'identité maximale est de n liens. On retrouve ainsi, sous une forme plus générale, le nombre moyen de messages échangés par l'algorithme :

$$E[M_n] = \sum_{i=1}^{n-1} E[T_i] + n = n \sum_{i=1}^{n} \frac{1}{n - i + 1} = nH_n.$$

De surcroît, les v.a. T_i ($1 \leq i \leq n - 1$) sont indépendantes et de même loi hypergéométrique négative : leur variance a pour expression

$$\mathrm{var}(T_i) = n \times \frac{(i-1)(n-i)}{(n-i+1)^2(n-i+2)} .$$

Comme les n messages envoyés par le processus d'identité maximale ne contribuent pas à la variance de M_n, on a par conséquent,

$$\mathrm{var}(M_n) = n \times \sum_{i=1}^{n-1} \frac{(i-1)(n-i)}{(n-i+1)^2(n-i+2)} . \qquad (1)$$

Enfin, au pire comme en moyenne, la taille maximale d'un message est en $O(\lg id^*)$, où id^* est l'identité maximale des processus, *i.e.* en $O(\lg n)$ si l'on convient que les identités sont dans $[n]$. La complexité moyenne en bits est donc en $O(n \lg n \lg id^*)$. ∎

3.3.3. *Optimalité en moyenne de l'algorithme*

La démonstration d'optimalité ci-dessous utilise des résultats simples de combinatoire des permutations dans la trame de la démonstration de [PACHL, KORACH,ROTEM-84] pour les anneaux bidirectionnels (voir aussi [BODLAENDER-91] et [TEL-94] où l'optimalité en messages est démontrée dans un cadre plus général).

Les hypothèses fondamentales sont celles du § 3.1 : on considére des anneaux unidirectionnels dont la taille n est inconnue des processus. Notons que, les anneaux étant asynchrone, la connaissance de n n'a, de toute façon, aucune influence sur l'optimalité en moyenne de l'algorithme. Sans perte de généralité, on suppose de plus que l'ensemble I des identités (distinctes) est fini et que $|I| = n$. Nous allons montrer que nH_n représente une borne inférieure de la complexité moyenne en messages pour tout algorithme d'élection d'extremum sur un anneau unidirectionnel asynchrone, *i.e.* de la classe ($\mathfrak{E}$). Soit A un algorithme de la classe ($\mathfrak{E}$), on note $C_A(I)$ le nombre total de messages échangés par A sur les anneaux étiquetés par les suites $\sigma \in I$, et $\overline{C_A(I)}$ la complexité moyenne en messages de A sur tous les anneaux étiquetés par des identités de I.

Soit donc un anneau unidirectionnel étiqueté par $s = \langle id_1\ id_2 \cdots id_n \rangle \in I$, c'est-à-dire par une permutation de I : $\sigma = \langle \sigma_1 \cdots \sigma_n \rangle \in \mathfrak{S}(I) = \mathfrak{S}_n$, où $\sigma_i \in I$ $(1 \le i \le n)$. Il existe $n!$ permutation dans $\mathfrak{S}_n$, mais seulement $(n-1)!$ manières distinctes d'étiqueter un anneau de taille n avec les σ_i $(1 \le i \le n)$; tout étiquetage d'un anneau de taille n correspond en effet à une *permutation circulaire de I*.

Définition 2 *Un n-cycle γ de I est une permutation dont les éléments mobiles, en nombre n, sont de la forme a, $\gamma(a)$, $\gamma^2(a)$, ..., $\gamma^{n-1}(a)$ ($\gamma^n(a) = a$), où $a \in I$.*

On note $\mathfrak{C}(I) = \mathfrak{C}_n \subset \mathfrak{S}_n$ l'ensemble de toutes les permutations circulaires de I, c'est-à-dire les n-cycles de I : $|\mathfrak{C}_n| = (n-1)!$

Un anneau étiqueté par une permutation $\sigma = \langle \sigma_1 \cdots \sigma_n \rangle$ de I est appelé un *σ-anneau*. Si τ est une permutation circulaire de σ, tout τ-anneau est évidemment encore « le même anneau » que le σ-anneau initial. Dans la suite, la notion de *préfixe* d'une permutation est utilisée dans son sens habituel : soient σ et τ deux permutations de $\mathfrak{S}_n$, τ est un préfixe de σ s'il existe une permutation π telle que $\sigma = \tau\,\pi$.

Selon les hypothèses faites précédemment, le nombre de messages échangés par A ne dépend pas des délais relatifs de transmission : les n processus sont supposés actifs simultanément au temps $t = 1$ — ce cas correspond au maximum de messages échangés, ce qui répond bien aux besoins du calcul d'une borne inférieure — et les délais de transmission des messages le long des liens de l'anneau sont, par hypothèse, d'une unité de temps. L'envoi ou non d'un message au temps t par un processus P_i ne dépend par conséquent que de sa propre identité *et* de celle des $t-1$ processus qui le précédent sur l'anneau, *i.e.* des processus situés *à distance au plus t* de P_i.

Théorème 2 *Pour tout algorithme A d'élection par calcul d'extremum sur un anneau unidirectionnel (classe ($\mathfrak{E}$)), et pour tout ensemble I de n identités,* $\overline{C_A(I)} \geq nH_n$.

Démonstration. Soit A un algorithme quelconque de la classe ($\mathfrak{E}$). Pour calculer une borne inférieure de $\overline{C_A(I)}$, il faut faire l'hypothèse que les $(n-1)!$ anneaux possibles étiquetés par des permutations de $\mathfrak{S}_n$ sont *équiprobables*. La moyenne sur $C_A(I)$ est alors calculée sur l'ensemble des permutations circulaires des identités. Soit donc $C_A(I)$ le nombre total de messages envoyés par A sur un anneau, pour $1 \leq t \leq n-1$ fixé.

$$C_A(I) \geq \sum_{\gamma \in \mathfrak{C}_n} |\{\text{préfixes } \pi \text{ de } \gamma \in \mathfrak{C}_n \text{ tels que } |\pi| = t\}| \times |\{\text{maxima de gauche à droite dans } \gamma\}|.$$

Or, pour t fixé, $\quad |\{\text{préfixes } \pi \text{ de } \gamma \in \mathfrak{C}_n \text{ tels que } |\pi| = t\}| = n,$

donc, $$C_A(I) \geq n \sum_{\gamma \in \mathfrak{S}_n} |\{\text{maxima de gauche à droite dans } \gamma\}|.$$

Par ailleurs, $\gamma = \langle a, \gamma(a), \gamma^2(a), \ldots, \gamma^{n-1}(a)\rangle$ (*cf.* définition 2), et le nombre moyen de maxima de gauche à droite dans la permutation $\gamma \in \mathfrak{S}_n$ est, comme on le sait, le $n^{\text{ième}}$ nombre harmonique H_n. Par conséquent, en effectuant la moyenne sur les $(n-1)!$ permutations circulaires de I,

$$\overline{C_A(I)} \geq \frac{1}{(n-1)!} \times n H_n |\mathfrak{S}_n|.$$

$$\overline{C_A(I)} \geq nH_n$$

et donc $$(\forall A \text{ dans } (\mathcal{E})) \quad \overline{C_A(I)} \geq nH_n. \quad \blacksquare$$

Théorème 3 *Si l'ensemble I possède n éléments, $\min_A \overline{C_A(I)} = nH_n$, où le minimum est pris sur tous les algorithmes A de la classe $(\mathcal{E})$.*

Démonstration. D'après le théorème précédent, $\min_A \overline{C_A(I)} \geq nH_n$; comme il existe un algorithme A_0 (Chang-Roberts) de la classe $(\mathcal{E})$ qui atteint cette borne : $\overline{C_{A_0}(I)} = nH_n$ (inauguration non comprise), le théorème est démontré. $\blacksquare$

La complexité en messages de l'algorithme de Chang-Roberts étant en moyenne $nH_n + O(n)$, l'algorithme est donc bien *en moyenne optimal* dans $(\mathcal{E})$ en nombre de messages, en $\Theta(n \lg n)$.

3.3.4. *Comportement de l'algorithme*

Déterminons maintenant la probabilité pour l'algorithme de Chang-Roberts de s'éloigner de sa valeur moyenne optimale lorsque n croît, c'est-à-dire la probabilité de « larges déviations » pour l'algorithme. Avec les mêmes notations que ci-dessus, on a la

Proposition 1 *Lorsque $n \to \infty$, pour tout $\varepsilon > 0$,*

$$\lim_{n \to \infty} \Pr\{M_n \leq (1+\varepsilon)\mathrm{E}[M_n]\} = 1.$$

Démonstration. Voici deux démonstrations de la proposition 1 qui illustrent bien la différence de vitesse de la convergence en probabilité selon les outils probabilistes utilisés : l'inégalité de Bienaymé-Tchebychev d'abord, puis celle de Tchernov ensuite.

i) Première inégalité (Bienaymé-Tchebychev)

D'après la valeur de var(M_n) (formule (1) du § 3.3.2),

$$\mathrm{var}(M_n) = n \times \sum_{i=1}^{n-1} \frac{(i-1)(n-i)}{(n-i+1)^2(n-i+2)} ,$$

$$\mathrm{var}(M_n) = n \times \sum_{j=2}^{n} \frac{(j-1)(n-j)}{j^2(j-1)} < n \times \sum_{j=2}^{n} \frac{(n-j)}{(j+1)^2} ,$$

donc $$\mathrm{var}(M_n) < n \times \sum_{j\geq 3} n/j^2 = n^2(\pi^2/6 - 5/4) < n^2 \times \pi^2/6.$$

Considérons l'inégalité de Bienaymé-Tchebychev :

$$\Pr\{M_n - \mathrm{E}[M_n] \geq \delta\, \mathrm{E}[M_n]\} \leq \mathrm{var}(M_n)/\delta^2 \mathrm{E}[M_n]^2 \leq \frac{n^2 \times \pi^2/6}{\delta^2 \mathrm{E}[M_n]^2} ,$$

soit $$(\forall \delta > 0) \quad \Pr\{M_n \geq (1+\delta)\mathrm{E}[M_n]\} \leq \frac{\pi^2/6}{\delta^2 {H_n}^2} . \qquad (2)$$

Le membre de droite de l'inéquation étant en $O((\ln n)^{-2})$ lorsque $n \to \infty$, pour tout $\delta > 0$, le nombre total de messages M_n transmis par l'algorithme de Chang-Roberts vérifie donc la relation

$$\Pr\{M_n < (1+\delta) n \ln n\} > 1 - O((\ln n)^{-2}),$$

et $$\lim_{n\to\infty} \Pr\{M_n < (1+\delta)\mathrm{E}[M_n]\} = 1. \qquad \blacksquare$$

ii) Seconde inégalité (Tchernov)

L'inégalité de Tchernov donne

$$(\forall \varepsilon > 0) \quad \Pr\{M_n \geq (1+\varepsilon)\mu\} \leq \exp(-\varepsilon^2 \mu/3).$$

Soit $(\forall \varepsilon > 0)\ \Pr\{M_n \geq (1+\varepsilon)\mu\} \leq \exp(-\varepsilon^2 n \ln n/3)$. (3)

Le membre de droite de l'inéquation étant en $O(\exp(-n \ln n))$ lorsque $n \to \infty$, pour tout $\varepsilon > 0$, le nombre total de messages M_n transmis par l'algorithme de Chang-Roberts vérifie donc la relation

$$\Pr\{M_n < (1+\varepsilon) n \ln n\} > 1 - O(\exp(-n \ln n)),$$

et $$\lim_{n\to\infty} \Pr\{M_n < (1+\varepsilon)\mathrm{E}[M_n]\} = 1.$$ ∎

Bien que les deux démonstration conduisent au même résultat du point de vue de la convergence en probabilité, il est clair que la *vitesse* de convergence obtenue par l'inégalité (3) (Tchernov) utilisée en *ii)* est sans commune mesure avec l'inégalité (2) en *i)* : en $O(\exp(-n \ln n))$ d'une part, en $O((\ln n)^{-2})$ de l'autre.

Ces résultats sont réellement significatifs du comportement en moyenne de l'algorithme de Chang-Roberts et prennent toute leur valeur par comparaison aux meilleurs algorithmes d'élection de la classe ($\mathcal{E}$) *dans le pire des cas*, c'est-à-dire aux majorants de [DOLEV,KLAWE,RODEH-82] et [HIGHAM,PRZYTYCKA-93] : 1,956 $n \ln n$ et 1,833 $n \ln n$, respectivement (*cf.* remarque du § 3.3.1).

Autrement dit, on peut établir à partir de l'inégalité (3) (Tchernov) que la probabilité que l'algorithme de Chang-Roberts ne dépasse pas 1,956 $n \ln n$ est supérieure à 0,81 sur des anneaux a 4 processus, à 0,95 pour $n = 6$ et à 0,99 pour $n = 8$ — on atteint 0,999 pour seulement $n = 10$. En revanche, le même calcul effectué avec l'inégalité (2) (Bienaymé-Tchebychev) nécessite des anneaux à 500 processus pour atteindre une probabilité supérieure à 0,95... et $n = 10^7$ pour 0,99. Les mêmes calculs effectués pour le meilleur majorant actuel, 1,833 $n \ln n$, donnent des résultat concordants : avec l'inégalité de Tchernov, la probabilité que l'algorithme de Chang-Roberts ne dépasse pas 1,833 $n \ln n$ est supérieure à 0,95 sur des anneaux a 7 processus et à 0,99 pour $n = 10$ — on atteint 0,999 pour seulement $n = 12$ —, tandis que l'inégalité de Bienaymé-Tchebychev nécessite des anneaux à 1000 processus pour une probabilité supérieure à 0,95... et $n = 10^7$ pour 0,99.

L'algorithme de Chang-Roberts a donc un comportement de très loin supérieur aux estimations indiquées habituellement par les auteurs à partir de l'inégalité de Tchebychev.

En résumé, la convergence en probabilité du nombre de messages de l'algorithme de Chang-Roberts vers sa valeur moyenne est très rapide et, en moyenne, cet algorithme est remarquablement performant. Cette vitesse de convergence ne peut cependant pas être constatée si l'on utilise des inégalités trop « faibles », comme celles de Tchebychev et Bienaymé-Tchebychev. La différence d'efficacité constatée ici entre outils probabilistes est particulièrement bien mise en évidence par les quelques valeurs consignées ci-dessus qui, en un sens, sont moins

significatives du comportement de l'algorithme lui-même que du très grand écart de puissance entre les inégalités utilisées.

3.3.5. *Complexité en temps de l'algorithme*

Le « temps idéal » d'exécution de l'algorithme, $\tilde{T}(n)$, vaut $2n - 1$ si tous les processus s'activent en même temps. En revanche, si un seul processus déclenche l'algorithme en passant seul dans l'état *actif* et qu'il s'agit du voisin « de droite » du processus d'identité maximale, nous sommes dans le pire des cas et

$$\tilde{T}_{\max}(n) = (n-1) + n + (n-1) = 3n - 2 = \Theta(n). \quad \blacksquare$$

Remarques

1. Dans l'algorithme de Chang-Roberts, le nombre *moyen* de « grands » messages est évidemment inférieur lorsqu'on élit le processus d'identité minimale. Dans les algorithmes de ce type, un message contient généralement une *valeur* d'identité de processus sur l'anneau considéré. Sans hypothèse de synchronicité, le nombre total de bits échangés est donc au moins égal au produit du nombre minimal de messages transmis $C_{\min}(n) = \Omega(n \lg n)$ par $\Omega(\lg id_*)$, où id_* est l'identité minimale sur l'anneau, soit en $\Omega(n \lg n \lg id_*)$.

2. Notons que le principe de l'extinction sélective qui est à la base de l'algorithme de Chang-Roberts est facilement généralisable à de nombreux algorithmes d'élection dans des réseaux quelconques (avec identités ou anonymes).

Il existe une variante *bidirectionnelle* de l'algorithme de Chang-Roberts, sa complexité moyenne en messages est $\sqrt{2}/2\, nH_n + O(n)$ (*cf.* [LAVAULT-90]). Elle est donc *strictement plus performante en moyenne* que l'algorithme original de Chang-Roberts, les constantes devant nH_n étant respectivement $1/\sqrt{2}$ et 1. De surcroît, cette variante est meilleure, en moyenne, *avec* ou *sans* le sens général de la direction sur l'anneau bidirectionnel asynchrone considéré : la connaissance générale de la direction est sans influence sur la complexité de l'élection sur les anneaux. Dans des réseaux quelconques en revanche, l'information structurelle permet de diminuer considérablement la complexité en communication de nombreux algorithmes.

En guise de brève conclusion à cette première partie, il faut souligner que la complexité en communication de l'élection sur un anneau asynchrone avec identités n'est influencée par presque aucune hypothèse ou information structurelle supplémentaire. Que l'anneau soit unidirectionnel ou bidirectionnel, que sa taille soit connue ou non des processus et qu'on considère la complexité au pire ou en moyenne, celle-ci reste toujours en $\Theta(n \lg n)$. La seule contrainte est de considérer un anneau *asynchrone* ; dans le cas synchrone en effet, la complexité de l'élection sur un anneau est linéaire.

4. L'élection sur des anneau anonymes

Nous avons vu précédemment que l'intérêt d'identités dans les réseaux va bien au-delà de la simple utilisation pratique pour l'échange de messages entre processus : les identités sont *comparables* et peuvent donc servir à briser la symétrie. C'est Dana Angluin [ANGLUIN-80] qui étudia la première les possibilités de calculs dans des réseaux anonymes et établit que ces derniers sont strictement moins puissants que leurs homologues déterministes : les réseaux centralisés et les réseaux avec identités peuvent en effet simuler les algorithmes distribués anonymes. Alon Itai et Michael Rodeh montrèrent dès 1981 [ITAI,RODEH-90] qu'en revanche, aucun algorithme distribué déterministe ne peut résoudre le problème de l'élection sur des anneaux anonymes (*cf.* § 4.1). Il est donc possible d'augmenter les possibilités des réseaux anonymes grâce au non-déterminisme, et aux algorithmes probabilistes en particulier.

4.1. *Les algorithmes probabilistes*

Dans un calcul réparti « classique », les transitions ne sont généralement pas définies de manière unique par l'état (global) du calcul. Un algorithme distribué réalise donc une *classe d'exécution possibles* plutôt qu'une exécution unique. L'important dans la modélisation probabiliste, c'est que l'introduction du stochastique engendre un *second* degré de non-déterminisme au sein des systèmes distribués, le système distribué standard asynchrone $\mathcal{S}$ décrit au § 2.1 étant déjà par essence un modèle non déterministe. Un algorithme distribué est déterministe si, dans chaque exécution potentielle de sa classe d'exécutions possibles, les processus terminent. Un algorithme probabiliste en revanche n'est pas tenu de réaliser une telle terminaison, mais seulement que les processus terminent avec une *grande probabilité* — 1 pour un algorithme de Las Vegas — dans *chaque* exécution de l'algorithme.

Dans l'analyse des algorithmes probabilistes, on considère toujours le pire des cas du point de vue du non-déterminisme induit par le système, exactement comme dans le cas déterministe, mais on calcule la moyenne *sur tous les choix aléatoire* de l'algorithme. Ainsi, bien que les algorithmes probabilistes aient le plus souvent des exécutions en temps *infini* ou *non borné* — et c'est d'ailleurs toujours le cas pour les algorithmes de Las Vegas —, un algorithme est néanmoins considéré comme correct si la probabilité d'une telle exécution est nulle.

Dans ces conditions, la terminaison n'est pas nécessairement du même type que pour les algorithmes distribués classiques : elle est dite « par processus » si elle s'effectue *avec* détection de la terminaison de l'algorithme, ou « par messages » si elle a lieu *sans* détection de la terminaison de l'algorithme (pour plus de détails, voir, entre autres, [ANGLUIN-80], [MATIAS,AFEK-89], [ITAI,RODEH-90], [LAVALLÉE,LAVAULT-91], [TEL-91,94] et le chapitre 5).

4.2. *Théorèmes d'impossibilité*

Les résultats fondamentaux qui suivent établissent l'impossibilité d'élire un processus de manière déterministe sur un anneau anonyme et la nécessité d'une connaissance générale de la taille n de l'anneau pour résoudre ce problème. Il faut noter que, comme dans l'algorithme analysé au § 4.3, les processus peuvent ne posséder qu'une connaissance partielle de n, par l'intermédiaire de son intervalle d'appartenance ou de ses bornes : un minorant N et un majorant du type kN (k et N entiers fixés connus des processus), *i.e.* $n \in [N,kN-1]$.

Théorème 4 [ANGLUIN-80] *Sur un anneau anonyme de taille n, aucun protocole déterministe ne peut résoudre le problème de l'élection — que n soit ou non connu des processus.*

La démonstration utilise le concept de *configurations symétriques* : il existe un calcul qui commence dans une configurations symétrique et atteint à nouveau une configurations symétrique toutes les n étapes de l'algorithme. L'impossibilité d'une solution déterministe se déduit alors du fait qu'aucun algorithme distribué correct ne peut terminer dans une configurations symétrique.

Théorème 5 [ITAI,RODEH-90] *Aucun algorithme probabiliste asynchrone ε-correct ne peut calculer la taille d'un anneau avec détection de terminaison.*

Un algorithme probabiliste de Monte Carlo est dit « ε-correct » s'il donne une solution correcte avec une probabilité d'au moins ε, $1/2 < \varepsilon < 1$, quelle que soit la donnée.

Ce second théorème d'impossibilité peut cependant se formuler de manière moins « négative » :

Théorème 6 [ITAI,RODEH-90] *Étant donné un paramètre $\varepsilon > 0$ quelconque et quelle que soit la taille n de l'anneau, il est possible de déterminer exactement n avec une probabilité d'erreur inférieure à ε grâce à un algorithme probabiliste de Monte Carlo.*

4.3. *Un algorithme d'élection sur un anneau anonyme unidirectionnel asynchrone*

L'algorithme A_1 présenté dans [ITAI,RODEH-90] est fondé sur le principe dit de « *sélection et vérification* » et utilise la même technique que l'algorithme d'élection sur des anneaux unidirectionnels asynchrones de [DOLEV,KLAWE,RODEH-82] ; sa complexité moyenne en bits en $O(n \lg n)$. Le principe de sélection et vérification est facilement généralisable à des réseaux anonymes quelconques.

4.3.1. *Présentation de l'algorithme* A_1

L'algorithme A_1 est donc constitué d'une phase de sélection suivie d'une phase de vérification. Initialement tous les processus sont actifs, durant une phase courante k, un certain nombre d'entre eux peuvent devenir inactifs, à la suite de quoi ils se contentent de relayer les messages en transit. Chaque processus actif effectue donc r fois les deux pas *(a)* et *(b)* qui suivent, en moins qu'ils ne passent dans l'état inactif avant la $r^{\text{ième}}$ exécution.

Pour tout processus P_i ;

(a) Tirer (sur {0,1}) aléatoirement, uniformément et indépendamment 0 ou 1 avec probabilité 1/2 ; envoyer le choix du tirage au prochain processus actif sur l'anneau ;

(b) Le processus actif P_i devient inactif *ssi* P_i a tiré un 0 en *(a) et* son premier prédécesseur actif sur l'anneau a tiré un 1.

Lorsque l'algorithme A_1 est terminé, il reste *au moins* un processus actif, puisqu'un processus ne peut devenir inactif que s'il a tiré un 0 tandis qu'un autre processus tirait un 1. Mais l'algorithme décrit ci-dessus ne peut pas garantir qu'un unique processus est élu : il se peut que plusieurs processus restent actifs. Il faut donc introduire une *phase de vérification* fondée sur l'envoi par tout processus actif d'un compteur autour de l'anneau lui permettant de vérifier s'il est ou non le seul processus encore actif.

4.3.2. *Analyse de l'algorithme* A_1

La première partie de cette analyse est due à Micha Hofri [HOFRI-87]. Soit I_t le nombre de processus actifs devenus inactifs à l'issue de l'étape t de l'algorithme A_1. Nous noterons $S_t = \sum_{j=1}^{t} I_j$, $N_t = n - S_t$ le nombre de processus encore actifs à l'issue de l'étape t et $G_t(z) = \sum_{s\geq 0} \Pr\{S_t = s\} z^s$, la fonction génératrice de probabilité (*FGP*) de S_t.

Pour calculer la distribution de I_1, il faut d'abord remarquer que, pour que k processus deviennent inactifs à la première étape, les n tirages aléatoires de ces processus sur {0,1} doivent constituer $2k$ séquences (ou « *runs* ») de 0 et de 1 sur l'anneau, ou encore, k séquences de mots $\omega = 10$, ces séquences pouvant commencer en $2k$ positions. Un processus peut donc devenir inactif de $2\times\binom{n}{2k}$ manières distinctes, le facteur 2 provenant du fait que, les points frontières étant fixés dans les $2k$ séquences, il existe deux possibilités de tirage pour le processus P_0 : 0 ou 1. Comme par ailleurs les n processus peuvent tirer 0 ou 1 de 2^n manières possibles,

$$\Pr\{I_1 = k\} = \frac{1}{2^n}\, 2\times\binom{n}{2k} = 2^{-n+1}\binom{n}{2k}.$$

La *FGP* de $I_1 = S_1$ est alors

$$G_1(z) = \sum_{k\geq 0} \Pr\{I_1 = k\} z^k = 2^{-n} \sum_{k\geq 0} 2\times\binom{n}{2k} z^k = 2^{-n} \sum_{j\geq 0, j=2k} 2\times\binom{n}{j} (\sqrt{z})^j,$$

soit,
$$G_1(z) = 2^{-n}\left[\,(1+\sqrt{z})^n + (1-\sqrt{z})^n\,\right].$$

Les premier et second moments de I_1 sont donc

$$\mathrm{E}[I_1] = G_1'(1) = n/4, \quad (n > 1)$$
$$\mathrm{E}[I_1^2] = G_1''(1) + \mathrm{E}[I_1] = n(n+1)/16.$$

Tant que le nombre de processus encore actifs à l'issue de l'étape $t-1$ est strictement plus grand que 1, $N_{t-1} > 1$, les étapes successives de l'algorithme sont conditionnées indépendamment par le nombre de processus encore actifs à chaque étape. Les résultats obtenus pour les moments de I_1 peuvent alors être généralisés à I_t, à condition de remplacer n par $n - S_{t-1} = N_{t-1}$. Pour surmonter la difficulté inhérente au cas $N_{t-1} \leq 1$, on définit un processus auxiliaire, noté $\tilde{N}_t$, défini à partir de la relation

$$\mathrm{E}[\tilde{I}_t \mid \tilde{S}_{t-1} = s] = \frac{n-s}{4}\ .$$

Ainsi,
$$N_t = \begin{cases} \tilde{N}_t & \text{si } \tilde{N}_t > 1, \\ 1 & \text{sinon.} \end{cases}$$

Les deux v.a. N_t et $\tilde{N}_t$ sont donc égales lorsqu'il n'y a pas de problème et telles que $\tilde{N}_t \leq 1 \Rightarrow N_t = 1$. Ceci permet alors d'étudier le comportement de $\tilde{N}_t$ à la place de celui de N_t.

LEMME 2
$$\mathrm{E}[\tilde{S}_t] = n\left[\,1 - \left(\frac{3}{4}\right)^t\,\right], \tag{5}$$

Démonstration. $\mathrm{E}[\tilde{S}_t] = \mathrm{E}[\tilde{S}_{t-1} + \tilde{I}_t] = \mathrm{E}[\tilde{S}_{t-1}] + \mathrm{E}[\tilde{I}_t]$ et,

par définition, $$\mathrm{E}[\tilde{I}_t \mid \tilde{S}_{t-1}] = \frac{n - \tilde{S}_{t-1}}{4},$$

soit $$\mathrm{E}[\tilde{I}_t] = \sum_{i \geq 0} \Pr\{\tilde{S}_{t-1} = i\}\,(n-i)/4 = \frac{n - \mathrm{E}[\tilde{S}_{t-1}]}{4}.$$

On a donc la relation de récurrence suivante,

$$\mathrm{E}[\tilde{S}_t] = \mathrm{E}[\tilde{S}_{t-1}] + \frac{n - \mathrm{E}[\tilde{S}_{t-1}]}{4},$$

$$\mathrm{E}[\tilde{S}_t] = 3/4\ \mathrm{E}[\tilde{S}_{t-1}] + n/4,$$
$$\mathrm{E}[\tilde{S}_1] = \mathrm{E}[\tilde{I}_1] = n/4. \qquad (4)$$

D'où l'espérance de $\tilde{S}_t$,

$$\mathrm{E}[\tilde{S}_t] = n\left[\ 1 - \left(\frac{3}{4}\right)^t\ \right]. \qquad \blacksquare \qquad (5)$$

A partir du lemme 2 (égalité (5)), il est maintenant possible d'évaluer la probabilité d'avoir plus d'un processus encore actif après $t = r$ étapes : $\Pr\{N_r > 1\}\ldots$ et de prouver ainsi qu'elle est faible. Pour ce faire, plusieurs inégalités probabilistes sont à notre disposition, celles de Bienaymé-Tchebychev (ou Tchebychev), de Markov et de Tchernov, entre autres. L'inégalité de Bienaymé-Tchebychev nécessitant la variance de $\tilde{S}_t$ en plus de son espérance, nous ne l'utiliserons donc pas. Les deux lemmes qui suivent évaluent deux inégalités sur $\Pr\{N_t > 1\}$: le premier est fondé sur l'inégalité — peu puissante — de Markov et le second sur celle de Tchernov, déjà rencontrée précédemment.

LEMME 3 *N_t étant toujours défini comme le nombre de processus encore actifs à l'issue de l'étape t, on a l'inégalité*

$$\Pr\{N_t > 1\} \leq n\left(\frac{3}{4}\right)^t. \qquad (6)$$

Démonstration. Immédiate à partir de l'inégalité de Markov et de l'expression de $\mathrm{E}[\tilde{S}_t]$ dans la formule (5) du lemme 2. Par Markov, on a

$$\Pr\{N_t > 1\} \leq \mathrm{E}[N_t].$$

Or $E[N_t] = E[n - \tilde{S}_t] = n - E[\tilde{S}_t] = n - n\left[1 - \left(\frac{3}{4}\right)^t\right] = n\left(\frac{3}{4}\right)^t$, (7)

le lemme 3 est donc démontré. ∎

LEMME 4 *N_t étant toujours défini comme le nombre de processus encore actifs à l'issue de l'étape t, on a l'inégalité*

$$\Pr\{N_t > 1\} \leq e^{7/3} \exp\left(-\frac{1}{n}\left(\frac{4}{3}\right)^t - n\left(\frac{3}{4}\right)^t\right). \quad (8)$$

Démonstration. Reprenons l'expression de $E[N_t]$ évaluée ci-dessus en (7). D'après l'inégalité de Tchernov, on a, en posant $\mu = E[N_t]$,

$$\Pr\{N_t > (1 - \varepsilon)\mu\} \leq \exp(-\varepsilon^2\mu/3),$$

soit
$$\Pr\{N_t > 1\} \leq \exp\left(-\frac{(1-\mu)^2}{3\mu}\right),$$

$$\Pr\{N_t > 1\} \leq e^{1/3} \exp(2 - 1/\mu - \mu) = e^{7/3} \exp(-1/\mu - \mu).$$

Donc
$$\Pr\{N_t > 1\} \leq e^{7/3} \exp\left(-\frac{1}{n}\left(\frac{4}{3}\right)^t - n\left(\frac{3}{4}\right)^t\right). \quad ∎$$

Le théorème 7 ci-dessous s'en déduit, avec deux valeurs « possibles » pour le nombre moyen d'étapes de l'algorithme $r(n)$: l'une découle du lemme 3, inégalité (6) (de Markov), l'autre du lemme 4, inégalité (8) (de Tchernov)...

Théorème 7 *Soit $r(n)$ le nombre moyen d'étapes nécessaires pour que l'algorithme A_1 élise un processus sur un anneau de taille n connue, unidirectionnel, anonyme et asynchrone.*

α) Pour $r(n) > 2\log_{4/3}(n) = 6{,}95211899\ldots \ln n$, la complexité moyenne en bits de l'algorithme A_1 est $C_b(A_1,n) = O(n \lg n)$.

β) Plus précisément, pour $r(n) > \ln n$, la complexité moyenne en bits de l'algorithme A_1 est $C_b(A_1,n) = O(n \lg n)$.

Démonstration. α) C'est un corollaire du lemme 3, inégalité (6), qui précède. Pour $r(n)/\ln n > 2/\ln(4/3)$, $r(n)\cdot\ln(4/3) > 2\ln n$, donc $(4/3)^{r(n)} > n^2$ et $\left(\frac{3}{4}\right)^{r(n)} < n^{-2}$. Dans ces conditions, pour $t = r(n)$, $n\left(\frac{3}{4}\right)^t < n^{-1}$ et, par conséquent, $\Pr\{N_t > 1\} = o(n^{-1})$. C'est-à-dire que la probabilité pour que plus d'un seul processus soit encore actif après $r(n)$ étapes est en $o(1/n)$ en prenant $r(n) = 7\ln n$.

β) De même, comme corollaire du lemme 4, inégalité (8) : pour $r(n) > \ln n$, $(\frac{3}{4})^{r(n)} > (\frac{3}{4})^{\ln n} = n^{\ln(3/4)}$. Dans ces conditions, pour $t = r(n)$, on a $\exp(-n\,(3/4)^t) < \exp(-n^{1+\ln(3/4)})$ et donc $\Pr\{N_t > 1\} = o(\exp(-n^{3/4}))$. La probabilité pour que plus d'un seul processus soit encore actif après $r(n)$ étapes est ici en $o(\exp(-n^{3/4}))$ en prenant seulement $r(n) = \ln n$.

Dans l'un ou l'autre cas, exactement un bit est transmis sur chaque lien durant la phase de sélection, la complexité en bits de chaque étape de cette phase est donc n et la complexité totale en bits de la phase de sélection est $r \cdot n$, soit en $\Theta(n \lg n)$. La phase de vérification comporte a processus actifs, chacun d'entre eux envoyant un compteur de $\lg n$ bits qui parcourt une distance de n liens sur l'anneau : d'où un coût total de $a \cdot n \cdot \lg n$ bits. Le nombre moyen de processus actifs étant 1, le coût moyen de cette seconde phase de vérification est en $O(n \lg n)$. Comme par ailleurs la probabilité de n'avoir aucun élu après la phase de sélection est en $o(1/n)$, la complexité moyenne en bits des phases suivantes est donc bien négligeable. D'où la complexité moyenne en bits de l'algorithme A_1, $\overline{C_b(A_1,n)} = O(n \lg n)$. ∎

4.4. *Un algorithme d'élection sur un anneau anonyme synchrone de taille n telle que $N \leq n < 2N$*

Dans ce paragraphe, le modèle utilisé est celui d'un anneau unidirectionnel anonyme *synchrone*. L'algorithme distribué A_2, proposé dans [ITAI,RODEH-90], résout le problème de l'élection en terminant par messages — *sans* détection de la terminaison —, lorsque la taille n de l'anneau n'est *pas* connue *exactement*. Tous les processus ont une connaissance partielle de n par l'intermédiaire de son intervalle d'appartenance $[N, 2N-1]$, ou de ses bornes : un minorant N fixé quelconque et un majorant pris ici égal à $2N-1$ pour simplifier l'algorithme et son analyse. Dans la mesure ou n n'est pas connu précisément et ou les processus n'ont pas d'identité, ces derniers sont incapables de reconnaître leurs propres messages sur l'anneau. Malgré cette difficulté, nous allons utiliser au mieux les connaissances globales sur n possédées par les processus, en particulier le fait que tout message parcourant la distance $2N-1$ autour de l'anneau visite à coup sûr tous les processus et rencontre exactement une fois son initiateur. Notons qu'il est possible de choisir des hypothèses plus générales sur n, telles que $N < n \leq kN$, k et N étant deux entiers quelconques ($k > 2$ fixé) supposés connus des processus, et d'utiliser la propriété précédente, avec $k > 2$, pour trouver un algorithme distribué probabiliste d'élection : généralisé aux réseaux anonymes quelconques, ce type de problème admet une solution avec ou sans détection de la terminaison, selon les cas.

4.4.1. *Présentation de l'algorithme* A_2

Comme l'algorithme A_1 précédent, l'algorithme A_2 comporte deux phases fondamentales, une phase de sélection et une phase de vérification.

• *La phase de sélection* est itérée plusieurs fois, jusqu'au choix d'un ou plusieurs candidats à l'élection, chaque itération dure $2N - 1$ cycles d'horloge. En début de cycle de chaque itération, tout processus P_i choisit une identité id_i en effectuant un tirage aléatoire indépendant dans $\{0,1\}$ selon une loi de Bernoulli de probabilité $p = r/N$, r étant, comme dans l'algorithme A_1 un paramètre entier connu de tous les processus. Autrement dit, $\Pr\{\text{succès}\} = \Pr\{id_i = 1\} = r/N$ et $\Pr\{\text{échec}\} = \Pr\{id_i = 0\} = 1 - r/N$: un processus ayant tiré l'identité 1 est alors considéré comme *candidat*. Chaque candidat envoie ensuite un jeton (un bit) le long de l'anneau au premier cycle de l'itération et relaie ensuite tout autre jeton reçu ultérieurement. Les autres processus attendent la réception d'un jeton avant de le renvoyer. Si aucun processus n'a tiré l'identité 1 il n'y a pas de communication, à la fin de l'itération tous les processus ont fini par apprendre l'absence de candidat et la phase de sélection doit être réitérée. La phase de sélection se termine par une dernière itération qui comprend *au moins un* candidat.

• *La phase de vérification* comprend deux sous-phases.

Au cours de la première sous-phase, un certain nombre de processus savent si un unique candidat a été ou non choisi, les autres processus l'apprennent durant la seconde sous-phase. La première sous-phase de vérification est très semblable à la phase précédente de sélection : les candidats choisis dans la phase de sélection ont entrepris de communiquer en envoyant un jeton (d'un bit) au cours du premier cycle d'horloge, mais, à la différence de la phase de sélection, ils ne doivent maintenant relayer *aucun* jeton reçu ultérieurement ; en revanche, sans modification par rapport à la phase précédente, les autres processus ne renvoient un jeton qu'après réception. Cette sous-phase toute entière ne dure que $N - 1$ cycles d'horloge. Dans le cas où il n'y a qu'*un seul* candidat, il envoie un unique message au cours de cette sous-phase ; celle-ci durant $N - 1 < n$ cycles, aucun message n'a la possibilité de parcourir un tour complet de l'anneau, et donc, le candidat en question ne peut recevoir *aucun message* pendant cette sous-phase. En revanche, dans le cas où il y a plusieurs candidats — *i.e.* au moins deux —, il existe deux candidats situés à une distance inférieure ou égale à $N - 1$ l'un de l'autre sur l'anneau, puisque $n \leq 2N - 1$. Donc, lorsque l'un d'entre eux reçoit un message au cours de cette sous-phase, il provient nécessairement de l'autre candidat (situé à une distance inférieure ou égale à $N - 1$) et ce candidat récepteur sait qu'il existe au moins un autre candidat sur l'anneau. Ainsi, lorsqu'il existe plusieurs candidats à la fin de la première sous-phase, l'un d'entre eux au moins le sait — mais pas nécessairement tous. S'il n'existe cependant qu'un unique candidat, il est incapable de déterminer qu'il est le seul à la fin de la première sous-phase ; la raison d'être de la seconde sous-phase est précisément de mettre les candidats au courant de leur nombre exact sur l'anneau.

La seconde sous-phase de vérification dure $2N - 1$ cycles d'horloge et est semblable aux phases précédentes. Les candidats sachant qu'il existe plus d'un candi-

dat sur l'anneau ont entrepris l'échange de communication et ne relaient donc plus aucun jeton qui pourrait être reçu ultérieurement. Les processus non candidats restant se contentent au contraire de relayer tous les jetons qu'ils reçoivent. Par conséquent, aucun jeton ne peut être envoyé au cours de cette sous-phase *ssi* il reste un seul et unique candidat sur l'anneau. En testant l'absence de communication durant cette sous-phase, les processus peuvent tous repérer qu'il n'existe qu'un unique candidat et ce dernier devient le processus élu. Dans le cas où un jeton est envoyé au cours de cette seconde sous-phase, tous les processus le reçoivent et il existe donc plusieurs candidats ; l'algorithme A_2 tout entier est alors réappellé.

4.4.2. *Analyse de la complexité de l'algorithme* A_2

Pour commencer l'analyse de la complexité de l'algorithme A_2, notons tout d'abord que si aucun candidat n'est choisi au cours de la phase de sélection, l'absence de communication est totale. Ainsi, indépendamment du nombre d'itérations de la phase de sélection, cette phase nécessite exactement n bits de communication (n jetons) pour chaque appel à l'algorithme A_2. La première sous-phase de vérification requiert $N-1$ bits lorsqu'il reste un unique candidat et n bits sinon. La seconde sous-phase fonctionne toujours avec exactement n bits de communication. En résumé donc, lorsqu'un seul et unique candidat reste sur l'anneau, tout nouvel appel à l'algorithme nécessite $n+N-1$ bits et sinon, il lui faut moins de $3n$ bits. Le temps (*i.e.* le nombre de cycles d'horloge) d'une itération est de $3N-2+\lambda\,(2N-1)$, où λ est le nombre d'itérations de la phase de sélection dans une exécution de l'algorithme A_2. Notons par ailleurs μ le nombre d'appels (ou d'itérations) à l'algorithme A_2. Les valeurs moyennes de λ et de μ, *i.e.* les espérances des variables aléatoires λ et μ, notées $\bar{\lambda}$ et $\bar{\mu}$, ainsi que la probabilité de réitération de l'algorithme A_2, dépendent de r, N et n.

Notons $c = c(r,N,n)$ la v.a. représentant le nombre de candidats, elle suit une loi binomiale $\mathcal{B}(n,r/N)$ telle que

$$\Pr\{c=i\} = \binom{n}{i}\left(\frac{r}{N}\right)^i\left(1-\frac{r}{N}\right)^{n-i}.$$

Lemme 5 *Soit λ la v.a. définie comme le nombre d'itérations de la phase de sélection dans un appel à l'algorithme A_2. En notant $p = 1-\Pr\{\lambda=1\}$, on a*

$$\bar{\lambda} = \mathrm{E}[\lambda] = \frac{1}{1-p} = 1/\Pr\{\lambda=1\}$$

et

$$\bar{\mu} = \mathrm{E}[\mu] = 1/\Pr\{\mu=1\}.$$

Démonstration. Posons $p_k = \Pr\{\lambda=k\}$, la probabilité pour que le nombre d'itérations de la phase de sélection de A_2 soit égale à k. La suite p_k vérifie la relation de récurrence suivante :

$$p_k = \Pr\{\lambda = k\} = \Pr\{c > 0\} \times (\Pr\{c = 0\})^{k-1},$$
$$p_1 = \Pr\{c > 0\}.$$

En effet, la v.a. λ est exactement égale à k *ssi* il n'y a *aucun* candidat au cours des $k-1$ itérations précédentes et *au moins* 1 lors de la $k^{\text{ième}}$ itération. Soit, en fonction de p,

$$p_k = (1-p)p^{k-1} \quad (k \geq 2),$$
$$p_1 = 1 - p. \tag{9}$$

Notons alors $P(z) = \sum_{k \geq 1} p_k z^k$ la fonction génératrice de probabilité (*FGP*) de la suite p_k. La relation de récurrence (9) s'écrit, pour $k \geq 2$, $p_k = p^{k-1} - p^k$ et, en multipliant les deux membres par z^k puis en sommant, on a la *FGP* $P(z)$ en fonction de p,

$$P(z) = \sum_{k \geq 1} p^{k-1} z^k - \sum_{k \geq 1} p^k z^k$$

$$= \frac{1}{p}\left(\frac{1}{1-pz}\right) - \frac{1}{1-pz},$$

soit
$$P(z) = \left(\frac{1-p}{p}\right)\left(\frac{1}{1-pz}\right).$$

On en tire donc
$$\bar{\lambda} = \mathrm{E}[\lambda] = P'(1) = \frac{1}{1-p} = 1/\Pr\{\lambda = 1\}.$$

De même,
$$\bar{\mu} = \mathrm{E}[\mu] = 1/\Pr\{\mu = 1\}. \quad \blacksquare$$

Théorème 8 *Sur un anneau de taille n, unidirectionnel, anonyme et synchrone dont les processus ne connaissent n que par l'intermédiaire de bornes de la forme* $N \leq n < 2N$, *l'algorithme distribué* A_2 *résout le problème de l'élection sans détection de terminaison, avec une complexité moyenne en bits* $\overline{C_b(A_2,n)} = (3\bar{\mu} - 2)n + N - 1$ *et une complexité moyenne en temps* $\overline{T_{A_2}(n)} = \bar{\mu}(3N - 2 + \bar{\lambda}(2N-1))$, *où*

$$\bar{\lambda} < (1 - e^{-r})^{-1} \qquad \text{et} \qquad \bar{\mu} < 1/r\left[e^{2r} \exp\left(\frac{r^2}{N-r}\right) - 1\right],$$

r étant un paramètre connu de tous les processus tel que $0 < r < N$.

Démonstration. Les valeurs de $\overline{C_b(A_2,n)}$ et $\overline{T_{A_2}(n)}$ découlent immédiatement de l'analyse de la complexité de A_2 ci-dessus et s'expriment en fonction de $\overline{\lambda(r,N,n)}$, le nombre moyen d'itération de la phase de sélection pour une exécution de A_2, et $\overline{\mu(r,N,n)}$, le nombre moyen d'itérations de l'algorithme A_2 tout entier, respectivement, comme suit :

$$\overline{C_b(A_2,n)} = (3\bar{\mu} - 2)n + N - 1 \text{ et } \overline{T_{A_2}(n)} = \bar{\mu}\,(3N - 2 + \bar{\lambda}\,(2N - 1)).$$

Il reste à évaluer des majorants de $\bar{\lambda}$ et $\bar{\mu}$:

• *Majoration de* $\bar{\lambda}$

On a, $\Pr\{\text{une unique itération de sélection suffit}\} = \Pr\{\lambda = 1\}$

$$\Pr\{\lambda = 1\} = \Pr\{c > 0\} = 1 - \Pr\{c = 0\}$$

$$= 1 - \left(1 - \frac{r}{N}\right)^n \geq 1 - \left(1 - \frac{r}{N}\right)^N,$$

donc $\Pr\{\text{une unique itération de sélection suffit}\} > 1 - e^{-r}$.

Comme, d'après le lemme 5, $\bar{\lambda} = 1/\Pr\{\lambda = 1\}$, la majoration de $\bar{\lambda}$ est immédiate :

$$\bar{\lambda} < \frac{1}{1 - e^{-r}} . \tag{10}$$

• *Majoration de* $\bar{\mu}$

On a, $\Pr\{A_2 \text{ est exécuté en une fois}\} = \Pr\{\mu = 1\}$,

$$\Pr\{\mu = 1\} = \Pr\{c = 1 \mid c > 0\} = \frac{\Pr\{c = 1 \wedge c > 0\}}{\Pr\{c > 0\}}$$

$$= \frac{\Pr\{c = 1\}}{\Pr\{c > 0\}} = \frac{\binom{n}{1}(r/N)(1 - r/N)^{n-1}}{1 - (1 - r/N)^n}$$

$$= \frac{nr/N}{(1 - r/N)^{1-n}\,[1 - (1 - r/N)^n]} ,$$

$$\Pr\{A_2 \text{ est exécuté en une fois}\} = \frac{nr/N}{(1 - r/N)\,[(1 - r/N)^{-n} - 1]} .$$

Donc, comme $\bar{\mu}$, le nombre moyen d'itérations de l'algorithme A_2, est tel que $\bar{\mu} = 1/\Pr\{A_2 \text{ est exécuté en une fois}\}$, on peut tirer de l'égalité précédente un majorant de $\bar{\mu}$

$$\bar{\mu} = \frac{(1 - r/N)[(1 - r/N)^{-n} - 1]}{nr/N} = 1/r \left(\frac{N - r}{n} \right) [(1 - r/N)^{-n} - 1]$$

et $$\bar{\mu} < 1/r\,[(1 - r/N)^{-n} - 1], \qquad \text{car } N - r < n$$

$$\leq 1/r\,[(1 - r/N)^{-(2N-1)} - 1].$$

Donc, $$\bar{\mu} < 1/r\,[(1 - r/N)^{-2N} - 1]. \tag{11}$$

Or, pour tout entier $r < N$,

$$(1 - r/N)^{-N} = \exp(-N \ln(1 - r/N))$$

$$= \exp\Big(N \sum_{j \geq 1} (r/N)^j/j \Big)$$

$$= \exp\Big(r + N \sum_{j \geq 2} (r/N)^j/j \Big) < e^r \exp\Big(N/2 \sum_{j \geq 2} (r/N)^j\Big)$$

et, $(1 - r/N)^{-N} < e^r \exp(N/2\,(r/N)^2\,(1 - r/N)^{-1}) \leq e^r \exp\left(\frac{r^2}{2(N - r)}\right)$.

On obtient par conséquent

$$\bar{\mu} < 1/r\,[e^{2r} \exp\left(\frac{r^2}{N - r}\right) - 1]. \quad \blacksquare \tag{12}$$

L'algorithme A_2 est un bon exemple de compromis entre la complexité en communication et la complexité en temps. Considérons en effet l'inéquation (11) ci-

dessus pour n et N fixés ; lorsque r décroît en tendant vers 0 ($0 < r < N$), $\overline{\mu}(r)$ est une suite décroissante de r.

Or $\lim_{r \to 0} 1/r\,[(1 - r/N)^{-2N} - 1] = 2$, ce qu'on montre par un calcul très simple. La suite $\overline{\mu}(r)$ étant ainsi décroissante majorée par 2, elle est convergente de limite $L < 2$. En revanche, dans les mêmes conditions (n et N fixés), on montre que, dans l'inéquation (10), $(1 - e^{-r})^{-1}$ tend vers l'infini lorsque $r \to 0$, et qu'il en est donc de même pour la suite $\overline{\lambda}(r)$.

En résumé, on a donc pour n et N fixés et lorsque $r \to 0$, $\lim_{r \to 0} \overline{C_b(A_2,n)} < 4n + N - 1$, tandis qu'en même temps, $\lim_{r \to 0} \overline{T_{A_2}(n)} = +\infty$, ce qui est invraisemblable.

Corollaire 1 *Sur un anneau de taille n, unidirectionnel, anonyme et synchrone dont les processus ne connaissent n que par l'intermédiaire de bornes de la forme* $N \leq n < 2N$, *l'algorithme distribué* A_2 *termine par messages en fournissant* une solution **quasi-optimale** *au problème de l'élection, avec une complexité moyenne en bits* $\overline{C_b(A_2,n)} = 6n + N - 1$ *et une complexité moyenne en temps* $\overline{T_{A_2}(n)} = 19N$.

Démonstration. Reprenons les résultats du théorème 8, c'est-à-dire

$$\overline{C_b(A_2,n)} = (3\overline{\mu} - 2)n + N - 1 \qquad \text{et}$$
$$\overline{T_{A_2}(n)} = \overline{\mu}\,(3N - 2 + \overline{\lambda}\,(2N - 1)) = (2\overline{\lambda} + 3\overline{\mu})N - (\overline{\lambda} + 2\overline{\mu}),$$

avec les inégalités (10) et (12) ($0 < r < N$) :

$$\overline{\lambda} < (1 - e^{-r})^{-1} \qquad \text{et} \qquad \overline{\mu} < 1/r\,\left[e^{2r} \exp\left(\frac{r^2}{N - r}\right) - 1\right].$$

Le compromis temps-communication peut être optimisé en cherchant le minimum du produit $(3\overline{\mu} - 2)\times(2\overline{\lambda} + 3\overline{\mu})$ ou, plus précisément, le minimum du produit $(3M_\mu - 2)\times(2M_\lambda + 3M_\mu)$, $M_\lambda(r)$ et de $M_\mu(r)$ étant les majorants respectifs de $\overline{\lambda}$ et $\overline{\mu}$ définis plus loin. Choisissons en effet r, $0 < r < N$, le problème est trivialement résolu pour $N = 1$, puisqu'on a alors $1 \leq n < 2$. Sans perte de généralité, il nous suffit donc de considérer le cas où $N \geq 2$. Sachant que le majorant $M_\mu(r)$ de $\overline{\mu}$ est une fonction décroissante de N, on a

$$\overline{\mu} < 1/r\,\left[e^{2r} \exp\left(\frac{r^2}{2 - r}\right) - 1\right] = 1/r\,\left[\exp\left(\frac{r\,(r - 4)}{r - 2}\right) - 1\right].$$

Posons donc $M_\lambda(r) = (1 - e^{-r})^{-1}$ et $M_\mu(r) = 1/r\,[\exp(\frac{r(r-4)}{r-2}) - 1]$. Il s'agit de trouver la valeur de $r \in]0,N[$, $N > 1$, qui minimise le produit $P(r)$:

$$P(r) = (3M_\mu(r) - 2)(2M_\lambda(r) + 3M_\mu(r)),$$

$$P(r) = \left\{ \frac{3}{r}\left[\exp\left(\frac{r(r-4)}{r-2}\right) - 1\right] - 2 \right\} \times \left\{ \frac{3}{r}\left[\exp\left(\frac{r(r-4)}{r-2}\right) - 1\right] + \frac{2}{1-e^{-r}} \right\}. \qquad (13)$$

Le minimum de la fonction $P(r)$ en (13) est 111,215... et il est atteint pour $r = 0{,}201\ldots$, c'est-à-dire pour $r = 1/5$ à 10^{-3} près. C'est le meilleur compromis entre $\overline{C_b(A_2,n)}$ et $\overline{T_{A_2}(n)}$: la solution est ainsi *quasi-optimale* pour $r \simeq 1/5$. En choisissant $r = 1/5$, on a

$$M_\mu(r) = 5\left[e^{2/5}\exp\left(\frac{1/25}{2-1/5}\right) - 1\right] = 5\,(e^{19/45} - 1) \simeq 2{,}6267,$$

$$M_\lambda(r) < (1 - e^{-1/5})^{-1} = \frac{e^{1/5}}{e^{1/5} - 1} \simeq 5{,}5166.$$

Donc
$$3M_\mu(r) - 2 \simeq 7{,}88 - 2 \simeq 5{,}88,$$
$$2M_\lambda(r) + 3M_\mu(r) \simeq 7{,}88 + 11{,}0333 \simeq 18{,}9135$$
et
$$M_\lambda(r) + 2M_\mu(r) \simeq 10{,}77.$$

D'où les solutions quasi-optimales, $\overline{C_b(A_2,n)} = 6n + N - 1$ et $\overline{T_{A_2}(n)} = 19N$. ∎

Après l'élection d'un processus, il est facile de déterminer la taille n de l'anneau par l'envoi d'un jeton autour de celui-ci à partir d'un processus P_i, suivi d'un décompte du nombre de cycles d'horloge nécessaires au jeton pour parcourir l'anneau tout entier et revenir en P_i. Il suffit ensuite de faire parcourir l'anneau deux fois par le jeton pour transmettre l'information en question : n est alors la différence de cycles d'horloge écoulés entre le premier et le second parcours du jeton.

L'algorithme A_1 du § 4.3 peut être appliqué à nouveau à l'élection sur un anneau de taille n, unidirectionnel, anonyme et asynchrone, mais cette fois, comme précédemment pour l'algorithme A_2 du § 4.4, sous l'hypothèse d'une connaissance partielle de n par les processus, *i.e.* par l'intermédiaire de bornes de la forme

$N \leq n < 2N$. La seule différence avec l'algorithme A_1 tient ici encore à la *non connaissance exacte de n* par les processus : tout message de vérification doit parcourir la distance $2N - 1$ autour de l'anneau. Le processus où s'arrête ce message sait alors s'il existe plus d'un élu ou non — le message en question rencontre en effet un élu au moins trois fois — et c'est lui qui prévient ensuite les autres processus de la réalisation de l'élection. La complexité moyenne en communication est ainsi en $O(n)$ messages, ou encore en $O(n \lg N + (2N - 1)\lg N) = O(n \lg n)$ bits en moyenne. Il faut par ailleurs $O(n \lg n)$ bits additionnels pour évaluer la taille n de l'anneau.

5. Commentaires et discussion autour de l'élection sur les anneaux

À la fin du chapitre 3, nous avons brossé une rapide conclusion à la première partie. Pour résumer brièvement les résultats du § 4.2, soulignons que l'élection d'un processus unique par un algorithme déterministe n'est possible que sur un anneau avec identités. Sur un anneau anonyme, aucune élection déterministe n'est possible et les algorithmes distribués probabilistes ne résolvent le problème que si, et seulement si, la taille de l'anneau est connue des processus — pas nécessairement de manière exacte cependant. Il s'ensuit que même les algorithmes probabilistes s'avèrent incapables de calculer la taille d'un anneau anonyme avec probabilité 1. Par ailleurs, de nombreux problèmes relatifs à la terminaison des algorithmes distribués, par processus ou par messages, sont posés *de facto* par les algorithmes d'élection sur les anneaux.

En fait, absolument tout ce qui vient d'être dit ci-dessus est également vrai dans des réseaux *quelconques* : la topologie très simple de l'anneau et le problème classique de l'élection englobent presque toutes les difficultés rencontrées plus généralement en algorithmique distribuée. Des problèmes aussi importants que, par exemple, la puissance de la terminaison par messages et, en ce qui nous concerne ici, la quasi-totalité des méthodes d'analyse de complexité en moyenne des algorithmes distribués sont remarquablement illustrés par l'élection sur les anneaux. De même, ces algorithmes se généralisent presque tous facilement aux cas des réseaux quelconques : le principes de l'extinction sélective de [CHANG, ROBERTS-79] et/ou la méthode utilisée par [DOLEV, KLAWE,RODEH-82] sont ainsi très efficacement utilisés dans des algorithmes d'élection plus généraux (*cf.* [MATIAS,AFEK-89], [LAVALLÉE,LAVAULT-91], [TEL-91,94], etc.).

Au vu de la complexité des algorithmes d'élection A_1 et A_2 (§ 4.3 et 4.4), on pourrait conclure que la connaissance exacte de la taille n de l'anneau par les processus joue un rôle moins important que le fait d'être synchrone ou asynchrone. En effet, avec la connaissance *exacte* de n, la complexité en communication de l'élection sur des anneaux anonymes *asynchrones* est en moyenne en $O(n \lg n)$ bits (algorithme A_1), tandis qu'elle elle linéaire sur les anneaux anonymes *synchrones*, malgré une simple connaissance *partielle* de n : $n \in [N, 2N - 1]$ (algorithme A_2). Ce serait cependant oublier que l'algorithme synchrone A_2 termine seulement *par messages* (*sans* détection de terminaison) alors que l'algorithme asynchrone A_1 termine *correctement* par processus. Les nombreux résultats de complexité en communication de l'élection sur les anneaux anonymes confirment en fait les conclusions déjà formulées à propos de l'élection sur les anneaux avec identités (§ 3). Sur les anneaux ano-

nymes comme sur les anneaux avec identités, l'élection avec terminaison par processus est caractérisée par une grande stabilité de la complexité en communication ; celle-ci n'est influencée par presque aucune hypothèse ou information structurelle. Que l'anneau soit unidirectionnel ou bidirectionnel, synchrone ou asynchrone et que sa taille soit connue exactement ou partiellement des processus, la complexité moyenne en communication demeure en $\Theta(n \lg n)$ bits (*cf.* [MATIAS,AFEK-89], [LAVALLÉE,LAVAULT-91], [TEL-91,94], etc.).

Plus généralement en ce qui concerne la terminaison, précisons que, dans les réseaux anonymes de taille inconnue, la classe des fonctions calculables par des algorithmes qui terminent par messages est *strictement plus grande* que la classe des fonctions calculables par des algorithmes qui terminent par processus. L'inclusion large de la seconde classe dans la première découle du fait qu'un algorithme qui termine par processus termine également par messages. L'inclusion stricte est un résultat démontré dans [TEL-94]. De plus, l'utilisation d'une procédure de *détection de terminaison* permet à un algorithmes quelconque qui termine par messages de terminer par processus. Dans les réseaux anonymes dont la taille est connue, il est ainsi toujours possible d'obtenir une détection de la terminaison, alors que dans des réseaux anonymes de taille inconnue, il n'existe aucun algorithme avec détection de terminaison.

Les problèmes de terminaison dans les systèmes distribués sont bien résolus sur les anneaux et, par généralisation, souvent aussi dans les réseaux quelconques. En revanche, on comprend que les problèmes liés à l'information structurelle (sens de la direction, connaissance de la topologie, connaissance de la taille du réseau, de son diamètre, etc.) et à même d'améliorer très considérablement la complexité en communication des algorithmes distribués, soient évidemment moins bien cernés sur un simple anneau (voir [SANTORO,URRUTIA,ZAKS-86], [LAVAULT-87], [TEL-91,94], etc). De nombreuses questions restent ouvertes dans ce domaine de l'algorithmique distribuée.

Soulignons enfin que, malgré leur manque (tout relatif) de puissance, seuls les algorithmes distribués probabilistes s'avèrent capables de répondre aux difficultés intrinsèques de l'algorithmique distribuée dans son évolution actuelle vers des systèmes distribués à réseaux asynchrones et anonymes composés de processus séquentiels communicants totalement identiques, dont l'état initial est arbitraire et inconnu.

6. Bibliographie

ANGLUIN Dana (1980). Local and Global Properties in Networks of Processors, *Proc. 12th ACM Symp. on Theory of Computing*, Los Angeles 1980, p. 82-93.

BODLAENDER Hans L. (1991). New Lower bounds Techniques for Distributed Leader Finding and Other Problems on Rings of Processors, *Theoretical Computer Science 81*, 1991, p. 237-256.

BODLAENDER Hans L. & TEL Gerard (1990). Bit-Optimal Election in Synchronous Rings, *Information Processing Letters Vol. 36*, 1990, p. 53-56.

CHANG Edwards J. & ROBERTS Robert (1979). An Improved Algorithm for Decentralized Extrema-finding in Circular Configurations of Processes, *Comm. of the ACM Vol. 22, N° 5*, mai 1979, p. 281-283.

COMTET Louis (1970). *Analyse combinatoire*, Tomes I et II, Presses Universitaires de France, 1970.

DOLEV Dany, KLAWE Mike & RODEH Michael (1982). An $O(n \log n)$ Unidirectionnal Distributed Algorithm for Extrema Finding in a Circle, *Journal of Algorithms 3*, 1982, p. 245-260.

FELLER William (1968,1971). *An Introduction to Probability Theory and its Applications*, Vol. I et II, Wiley, 1968-1971.

FLAJOLET Philippe & ODLYZKO Andrew M. (1990). Singularity Analysis of Generating Functions, *SIAM Journal on Discrete Math. 3 (2)*, mai 1990, p. 216-240.

GRAHAM Ronald L., KNUTH Donald E. & PATASHNIK Oren (1990). *Concrete Mathematics. A Foundation for Computer Science*, Addison-Wesley 4ème Edition, 1990.

HIGHAM Lisa & PRZYTYCKA T. (1993). A Simple, Efficient Algorithm for Maximum Finding on Rings, *Proc. 7th Int. Workshop on Distributed Algorithms, Lecture Notes in Computer Science 725*, Springer-Verlag 1993, p. 249-263.

HOFRI Micha (1987). *Probabilistic Analysis of Algorithms*, Springer-Verlag, 1987.

ITAI Alon & RODEH Michael (1990). Symmetry Breaking in Distributed Networks, *Information and Computation 88 (1)*, p. 60-87.

JOHNSON Norman L. & KOTZ Samuel (1977). *Urn Models and their Application*, Wiley & Sons, 1977.

KNUTH Donald E. (1968,1969,1973). *The Art of Computer Programming*, Vol. 1, 2 et 3, Addison-Wesley, 1968, 1969 et 1973.

LAVALLÉE Ivan & LAVAULT Christian (1991). Spanning Tree Construction for Nameless Networks, *Proc. 4th Int. Workshop on Distributed Algorithms, Lecture Notes in Computer Science 486*, Springer-Verlag, 1991, p. 41-56.

LAVAULT Christian (1987). *Algorithmique et Complexité distribuées*, Thèse de Doctorat d'Etat, Université Paris XI, décembre 1987.

LAVAULT Christian (1990). Exact Average Complexity Values for Distributed Election on bidirectionnal Rings of Processors, *Theoretical Computer Science 73*, North-Holland 1990, p. 61-79.

LE LANN Gérard (1977). Distributed Systems — Towards a Formal Approach, *Information Processing'77*, North-Holland 1977, p. 155-160.

MATIAS Yossi & AFEK Yehuda (1989). Simple and Efficient Election Algorithms for Anonymous Networks, *Proc. 3rd Int. Workshop on Distributed Algorithms, Lecture Notes in Computer Science 392*, Springer-Verlag, 1989, p. 183-194.

MATTERN Friedemann (1989). Message Complexity of Simple Ring-based Election Algorithms — an Empirical Analysis, *Proc. 9th Int. Conf. Distributed Computing Systems*, 1989, p. 94-100.

PACHL Jan, KORACH Ephraïm & ROTEM Doron (1984). Lower Bounds for Distributed Maximum-finding, *Journal of the ACM, Vol. 31, N°4*, octobre 1984, p. 905-918.

RAYNAL Michel (1985). *Algorithmes distribués et Protocoles*, Eyrolles, 1985.

ROTEM Doron, KORACH Ephraïm & SANTORO Nicola (1987). Analysis of a Distributed Algorithm for Extrema Finding in a Ring, *Journal of Parallel and Distributed Computing 4*, 1987, p. 575-591.

SANTORO Nicola, URRUTIA Jorge & ZAKS Shmuel (1986). Sense of Direction and Communication Complexity in Distributed Networks, *Proc. of the 1st International Workshop on Distributed Algorithms* , Carleton Un. Press, 1986, p. 123-132.

TEL Gerard (1991). *Topics in Distributed Algorithms*, Cambridge Int. Series on Parallel Computation, Vol. 1, Cambridge University Press, 1991.

TEL Gerard (1994). *Introduction to Distributed Algorithms*, Cambridge Un. Press, 1994.

VITTER Jeffrey S. & FLAJOLET Philippe (1990). Average-Case Analysis of Algorithms and Data Structures, *in Algorithms and Complexity, Handbook of Theoretical Computer Science — Vol. A, Chap. 9*, éd. Jan Van Leeuwen, Elsevier & MIT Press, 1990, p. 431-524.

Unité de Recherche INRIA Rennes
IRISA, Campus Universitaire de Beaulieu 35042 RENNES Cedex (France)

Unité de Recherche INRIA Lorraine Technopôle de Nancy-Brabois - Campus Scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 VILLERS LES NANCY Cedex (France)
Unité de Recherche INRIA Rhône-Alpes 46, avenue Félix Viallet - 38031 GRENOBLE Cedex (France)
Unité de Recherche INRIA Rocquencourt Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)
Unité de Recherche INRIA Sophia Antipolis 2004, route des Lucioles - B.P. 93 - 06902 SOPHIA ANTIPOLIS Cedex (France)

EDITEUR
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 LE CHESNAY Cedex (France)

ISSN 0249 - 6399